研究ノート

# Maple を用いた量子化学入門教材の作成

## Development of Web-based Teaching Materials for Junior Students Using Maple and MOPAC System: An introduction to Hückel Molecular Orbital method for junior students

栗 原 照 夫*・上 村 拓 矢*
KURIHARA, Teruo; KAMIMURA, Takuya

大学の化学教育と高等学校化学教育との間で最も大きな違いの一つは量子化学に立脚する，原子・分子の理解であると考えられる。昨年，数式処理ソフト Maple を用いて水素原子の波動関数・電子密度等を視覚化した教材について報告したが，今回は $\pi$ 電子のみを取り扱う最も単純な Hückel 分子軌道法の教材を作成した。

## 1. Hückel 分子軌道法

$CH_2$=CH−CH=CH−CH=$CH_2$ のように，二重結合と単結合が交互にあるような構造を持っている化合物を共役化合物と呼ぶ。上はヘキサ−1, 3, 5−トリエンの例である。最も単純な共役ジエンである 1, 3−ブタジエン $CH_2$=CH−CH=$CH_2$ を構成する 4 個の炭素原子は全て。$sp^2$ 混成であり，各炭素原子はそれぞれ 3 個の $sp^2$ 混成軌道を用いて 3 個の C−C $\sigma$ 結合と 6 個の C−H $\sigma$ 結合を作り，これらの結合は同一平面内に位置している。各炭素原子上の，混成にあずからない 2p 軌道は 1 個ずつの p 電子をもち，この骨格平面に対して垂直方向に互いに平行に配置している。これらの電子を $\pi$ 電子といい，1, 3−ブタジエンの 4 個の $\pi$ 電子はある程度分子全体に分布しているのである。1, 3−ブタジエンの分子軌道を $\varphi$（ギリシア文字で phi）とすると，$\varphi$ は 4 個の炭素原子の 2p$\pi$ 原子軌道 $\chi$（chi）の 1 次結合で表すことができる。

$$\varphi = C_1\chi_1 + C_2\chi_2 + C_3\chi_3 + C_4\chi_4 \quad \cdots\cdots (1)$$

$C_1$, $C_2$, $C_3$, $C_4$ は分子軌道の係数である。1, 3−ブタジエンの場合，軌道は 4 軌道あり，$\pi$ 電子の

* 城西大学理学部化学科

数は 4 つである。分子軌道のエネルギー $\varepsilon$ は( 2 )式である。

$$\varepsilon = \frac{\int \varphi H \varphi d\tau}{\int \varphi^2 d\tau} \geq \varepsilon_0 \quad \cdots\cdots(2)$$

( 2 )式に基づいて最低エネルギーを与える係数を求めるためには，変分法を用い

$$\frac{\partial \varepsilon}{\partial C_i} = 0 \quad \cdots\cdots(3)$$

でなければならない。途中を省略すると，次の関係式が得られる。

$$\sum_j C_j (H_{ij} - \varepsilon S_{ij}) = 0 \quad \cdots\cdots(4)$$

この式を永年方程式と呼ぶ。従って( 5 )式を満足させるようにパラメーター $\varepsilon$ を決めればよい。

$$\left| H_{ij} - \varepsilon S_{ij} \right| = 0 \quad \cdots\cdots(5)$$

この式の導入は 2 年次あるいは 3 年次の講義でお目にかかれる。1, 3-ブタジエンの場合，次の 4 個の方程式が導かれる。

$$\begin{aligned}
&C_1(H_{11} - S_{11}\varepsilon) + C_2(H_{12} - S_{12}\varepsilon) + C_3(H_{13} - S_{13}\varepsilon) + C_4(H_{14} - S_{14}\varepsilon) = 0 \\
&C_1(H_{21} - S_{21}\varepsilon) + C_2(H_{22} - S_{22}\varepsilon) + C_3(H_{23} - S_{23}\varepsilon) + C_4(H_{24} - S_{24}\varepsilon) = 0 \\
&C_1(H_{31} - S_{31}\varepsilon) + C_2(H_{32} - S_{32}\varepsilon) + C_3(H_{33} - S_{33}\varepsilon) + C_4(H_{44} - S_{44}\varepsilon) = 0 \\
&C_1(H_{41} - S_{41}\varepsilon) + C_2(H_{42} - S_{42}\varepsilon) + C_3(H_{43} - S_{43}\varepsilon) + C_4(H_{44} - S_{44}\varepsilon) = 0
\end{aligned} \quad \cdots\cdots(6)$$

HMO 法では次の仮定がなされる。

⑴　重なり積分　$S_{ij} = \int \chi_i \chi_j d\tau$, $S_{ii} = 1$, $i \neq j$ のときは，すべての $S_{ij}$ に対して $S_{ij} = 0$ とする。

⑵　クーロン積分　$H_{ii} = \int \chi_i h \chi_i d\tau$ はすべて等しく，$H_{ii} = \alpha$ とする。

⑶　共鳴積分　$H_{ij} = \int \chi_i h \chi_j d\tau$, 原子軌道 $\chi_i$ と $\chi_j$ が隣接している場合はすべて $\beta$ に等しく，隣接していなければ，$H_{ij} = 0$ とおく。

HMO 法は上のような粗い近似を用いているが，$\pi$ 電子系化合物の性質，たとえば軌道エネルギー，$\pi$ 電子密度，電子親和力，イオン化ポテンシャル，電子スペクトル，共鳴エネルギーなどの物理化学的量をある程度，定量的に取り扱うことができる。又，Woodward-Hoffman 則に代表されるペリ環状反応の理解をより深めることも可能である。以上の近似を用いると( 6 )式は

$$\begin{aligned}
&C_1(\alpha - \varepsilon) + C_2\beta = 0 \\
&C_1\beta + C_2(\alpha - \varepsilon) + C_3\beta = 0 \\
&C_2\beta + C_3(\alpha - \varepsilon) + C_4\beta = 0 \\
&C_3\beta + C_4(\alpha - \varepsilon) = 0
\end{aligned} \quad \cdots\cdots(7)$$

と表現される。この式を$\beta$で割り，$(\alpha-\varepsilon)/\beta = -\lambda$と置き換えると

$$\begin{aligned} -C_1\lambda + C_2 &= 0 \\ C_1 - C_2\lambda + C_3 &= 0 \\ C_2 + C_3\lambda + C_4 &= 0 \\ C_3 - C_4\lambda &= 0 \end{aligned} \quad \cdots\cdots(8)$$

となる。別な表現をすると，(9)式になり，

$$\begin{aligned} C_1\lambda &= C_2 \\ C_2\lambda &= C_1 + C_3 \\ C_3\lambda &= C_2 + C_4 \\ C_4\lambda &= C_3 \end{aligned} \quad \cdots\cdots(9)$$

これに対応する行列式は

$$\begin{vmatrix} -\lambda & 1 & 0 & 0 \\ 1 & -\lambda & 1 & 0 \\ 0 & 1 & -\lambda & 1 \\ 0 & 0 & 1 & -\lambda \end{vmatrix} \quad \cdots\cdots(10)$$

であり，この式を解き，$\lambda$を求めれば$\varepsilon = \alpha + \lambda\beta$よりエネルギーが求まり，さらにこの$\lambda$と(11)式の規格化条件

$$C_1^{\,2} + C_2^{\,2} + C_3^{\,2} + C_4^{\,2} = 1 \quad \cdots\cdots(11)$$

より係数が求まる。この行列式の$\lambda$と係数Cを求める手順は固有値問題として化学の分野に良く現れる数学的手法であり，多様な解法が提出されている。今回の教材ではこの解法に数式処理ソフトMapleを用いた(10)式をMaple等で解く場合，$\lambda$を0とする。

その結果

$\varepsilon_4 = \alpha - 1.6180\beta \quad \varphi_4 = -0.3715\chi_1 + 0.6150\chi_2 - 0.6015\chi_3 + 0.3715\chi_4$

$\varepsilon_3 = \alpha - 0.6180\beta \quad \varphi_3 = 0.6015\chi_1 - 0.3715\chi_2 - 0.3715\chi_3 + 0.6015\chi_4$

$\varepsilon_2 = \alpha + 0.6180\beta \quad \varphi_2 = -0.6015\chi_1 - 0.3715\chi_2 + 0.3715\chi_3 + 0.6015\chi_4$

$\varepsilon_1 = \alpha + 1.6180\beta \quad \varphi_1 = 0.3715\chi_1 + 0.6015\chi_2 + 0.6015\chi_3 + 0.3715\chi_4$

が得られ，4個の$\pi$電子は1番安定な軌道に2個，2番目の軌道に2個入ることになる。

## 2. 永年方程式の組み立て方

式(９)は別の言い方をすると次のようである。1, 3-ブタジエンの炭素原子に番号を付ける。番号の付け方は任意であるが，永年方程式が解きやすく，又，分かりやすく番号付けを行う方が良い。

$C_1-C_2-C_3-C_4$（二重結合を省略する。以下同様）

$$\begin{aligned} C_1\lambda &= C_2 \\ C_2\lambda &= C_1+C_3 \\ C_3\lambda &= C_2+C_4 \\ C_4\lambda &= C_3 \end{aligned} \quad \cdots\cdots(9)$$

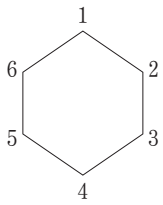
今，１番目の炭素 $C_1$ を考えると $C_1$ に $\lambda$ を掛け，$C_1$ に隣接する炭素原子は $C_2$ であるから，$C_1\lambda = C_2$ と書く。又，炭素 $C_2$ を考えると，$C_2$ に $\lambda$ を掛け，$C_2$ に隣接している炭素原子は $C_1$ と $C_3$ であるから，$C_2\lambda = C_1+C_3$ とする。以下同じ表現を $C_4$ 原子まで行う。この(９)式から(10)式の行列式を組み立てるには，$C_1\lambda = C_2$ は $C_1$ の係数を $-\lambda$ と考え，又，$C_2$ の係数を１と考えると $-\lambda \quad 1 \quad 0 \quad 0$ を得る。$C_2\lambda = C_1+C_3$ は $C_2$ の係数を $-\lambda$ と考え，$C_1$ と $C_3$ の係数もそれぞれ１と考える。以下，$C_3\lambda = C_2+C_4$, $C_4\lambda = C_3$ も同様に考えると，(10)式を組み立てることができる。

$$\begin{vmatrix} -\lambda & 1 & 0 & 0 \\ 1 & -\lambda & 1 & 0 \\ 0 & 1 & -\lambda & 1 \\ 0 & 0 & 1 & -\lambda \end{vmatrix} \quad \cdots\cdots(10)$$

もし，番号付けを $C_2-C_3-C_1-C_4$ とつけると

$$\begin{aligned} C_2\lambda &= C_3 \\ C_3\lambda &= C_1+C_2 \\ C_1\lambda &= C_3+C_4 \\ C_4\lambda &= C_1 \end{aligned} \quad \Rightarrow \quad \begin{vmatrix} 0 & -\lambda & 1 & 0 \\ 1 & 1 & -\lambda & 0 \\ -\lambda & 0 & 1 & 1 \\ 1 & 0 & 0 & -\lambda \end{vmatrix} = 0$$

となり，計算結果は同じであるが，はるかに(10)式の表現のほうが分かりやすい。もう１つベンゼンの例を示す。番号を次のようにつけた。




$$
\begin{aligned}
C_1\lambda &= C_2+C_6\\
C_2\lambda &= C_1+C_3\\
C_3\lambda &= C_2+C_4\\
C_4\lambda &= C_3+C_5\\
C_5\lambda &= C_4+C_6\\
C_6\lambda &= C_1+C_5
\end{aligned}
\Rightarrow
\begin{vmatrix}
-\lambda & 1 & 0 & 0 & 0 & 1\\
1 & -\lambda & 1 & 0 & 0 & 0\\
0 & 1 & -\lambda & 1 & 0 & 0\\
0 & 0 & 1 & -\lambda & 1 & 0\\
0 & 0 & 0 & 1 & -\lambda & 1\\
1 & 0 & 0 & 0 & 0 & -\lambda
\end{vmatrix} = 0
$$

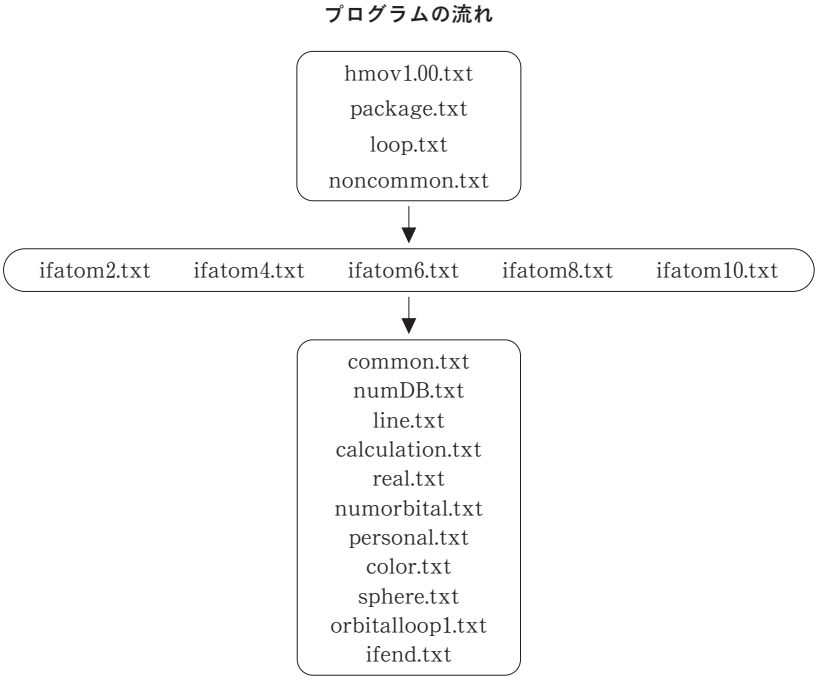
ベンゼンは 6 個の軌道で，$\pi$ 電子の数は $6\pi$ である。Maple でこの行列を入力するときは，前に述べたように $-\lambda$ を 0 とおく。

## 3. 分子軌道の可視化

数式処理ソフトは多数市販されているが，本学では数式処理ソフト Maple 10 が導入されており，Maple を用いた可視化を試みた。本教材は Maple 10 での動作確認を行っている。

### 3.1　教材の構成内容

本教材は複数のテキストファイルから構成されており，プログラム本体は説明文と最初のテキストファイルを読み込む文章しか表示されておらず，操作環境を分かりやすくした。全体のプログラムの流れは以下のようになる。

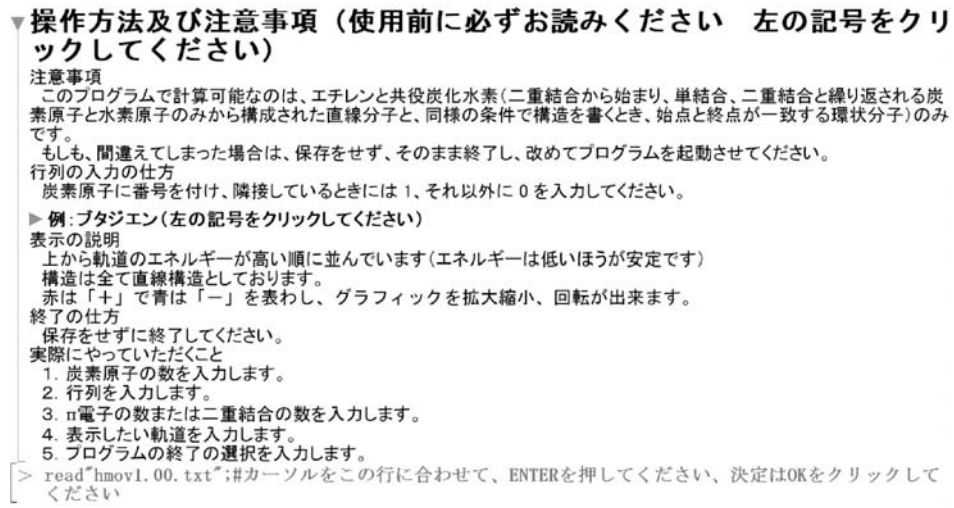


### 3.2　教材の実行方法

教材は指定された行で ENTER を押し実行すると対話形式で動作するようにし，Maple の入力形式に従い入力していくことで，その結果を表示するようにした。教材の実行方法を説明する。なお教材中でも説明してある。

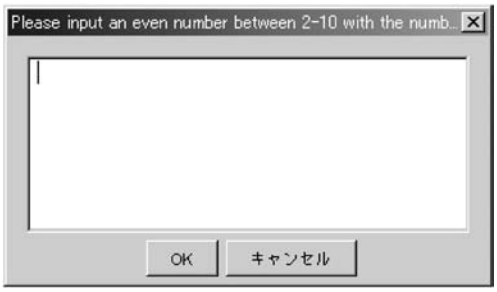
教材を起動させると


のように，表示される。図の左上の記号をクリックすること，次のように注意事項が表示される。




read″hmov1.00.txt″ と書かれている行にカーソルを合わせて，ENTER を押すとプログラムが実行する。すると，このように表示されるので炭素原子数を入力する。

炭素原子数を「2〜10」で入力してください

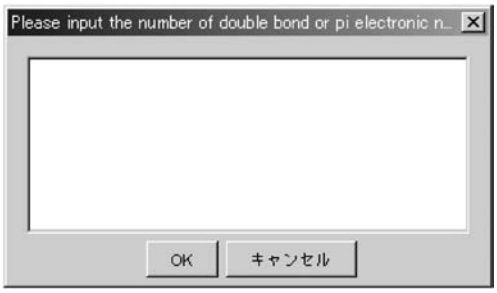


次の画面は行列式入力になり，行列式を入力する。図はブタジエンの入力例です。


行列式の入力が終了すると，次の画面のようになる。

炭素原子数は 4 です

二重結合の数または，$\pi$ 電子数を入力してください




二重結合の数，あるいは $\pi$ 電子数の数を入力すると，次のように表示される。

二重結合数は，2 です。行列は以下のようになります。

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

軌道のエネルギー

$\varepsilon_4 = \alpha - 1.6180\beta$

$\varepsilon_3 = \alpha - 0.61800\beta$

$\varepsilon_2 = \alpha + 0.61800\beta$

$\varepsilon_1 = \alpha + 1.6180\beta$

軌道は合計 4 あります。何番目の軌道を表示しますか？

行列式，軌道のエネルギーの一覧，軌道の数が表示され，希望する軌道の番号を入力する。例え

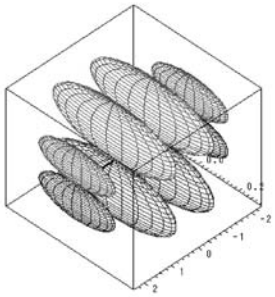
ばブタジエンの１番目の軌道を表示したいときは，１を入力する。

1番目の軌道を表示します。選択した軌道のエネルギーは

$$\varepsilon_1 = \alpha + 1.6180\beta$$

対応する分子軌道は次のようになります。

$$\varphi_1 = 0.37177\chi_1 + 0.60150\chi_2 + 0.60150\chi_3 + 0.37173\chi_4$$



選択した軌道のエネルギー，分子軌道，そのグラフィックが表示されます。

この教材では再実行をするか聞いてくるので，対応する数値を入力する。

### 3.3　教材のプログラム

プログラムの全体の流れは 3.1 に示したが，個別のプログラムを以下に示す。

```
restart:
Digits:=5:
#パッケージ読み込みと説明文
#パッケージの呼び出し
with(LinearAlgebra):
with(plots):
with(plottools):#ループ部分
totalloop:=1:
while totalloop=1 do
 #非共通部分
print(炭素原子数を「2〜10」の間で入力してください);
Acnt:=1:
while Acnt=1 do
#原子数の選択
ATOM:=readstat("Please input it between 10 from 2"):
```

```
 if ATOM＝2 then
  read"ifatom2.txt":
 elif ATOM＝3 then
  read"ifatom3.txt":
 elif ATOM＝4 then
with(Maplets[Elements]):
maplet1:＝Maplet(Window('title'＝"行列式", [
[TextField[TF[1][1]](3), TextField[TF[1][2]](3), TextField[TF[1][3]](3), TextField[TF[1]
[4]](3)],
[TextField[TF[2][1]](3), TextField[TF[2][2]](3), TextField[TF[2][3]](3), TextField[TF[2]
[4]](3)],
[TextField[TF[3][1]](3), TextField[TF[3][2]](3), TextField[TF[3][3]](3), TextField[TF[3]
[4]](3)],
[TextField[TF[4][1]](3), TextField[TF[4][2]](3), TextField[TF[4][3]](3), TextField[TF[4]
[4]](3)],
Button("OK", Shutdown([(TF[1][1], TF[1][2], TF[1][3], TF[1][4]), (TF[2][1], TF[2][2], TF
[2][3], TF[2][4]), (TF[3][1], TF[3][2], TF[3][3], TF[3][4]), (TF[4][1 ], TF[4][2], TF[4][3],
TF[4][4])]))
])):
A:＝Maplets[Display](maplet1);
＃1 次元を 2 次元に直し，実数化
for i from 1 to ATOM do
    for j from 1 to ATOM do
     A[i][j]:＝parse(A[ATOM*(j－1)＋i]):
    end do:
end do:
c:＝〈〈A[1][1], A[1][2], A[1][3], A[1][4]〉|〈A[2][1], A[2][2], A[2][3], A[2][4]〉|〈A[3][1],
A[3][2], A[3][3], A[3][4]〉|〈A[4][1], A[4][2], A[4][3], A[4][4]〉〉;
Acnt:＝0:
 elif ATOM＝5 then
  read"ifatom5.txt":
 elif ATOM＝6 then
  read"ifatom6.txt":
```

```
 elif ATOM＝7 then
  read"ifatom7.txt":
 elif ATOM＝8 then
  read"ifatom8.txt":
 elif ATOM＝9 then
  read"ifatom9.txt":
 elif ATOM＝10 then
  read"ifatom10.txt":
 else
  print(偶数を入力してください);
  print(炭素原子数を「2〜10」の間で入力してください);
 end if:
end do:
 ＃共通部分
 print(炭素原子数は，ATOM, です);
＃電子数の選択
DBcnt:＝1:
while DBcnt＝1 do
  ＃電子数入力
print(二重結合の数または，pi, 電子数を入力してください);
DB:＝readstat("Please input the number of double bond or pi electronic number of pi"):
if DB＝ATOM/2 then
 print(二重結合数は，DB, です);
    DBcnt:＝0:
elif DB〈＝ATOM＋2 then
 print(pi, 電子数は，DB, です);
 DBcnt:＝0:
elif DB〉＝ATOM－1 then
 print(pi, 電子数は，DB, です);
 DBcnt:＝0:
else
 print(それでは成り立ちません);
end if:
```

```
end do:
＃座標の設定
＃右側の座標[1][x]
l[1][1]:＝0.700:
l[1][2]:＝1.400:
l[1][3]:＝2.100:
l[1][4]:＝2.800:
l[1][5]:＝3.500:
l[1][6]:＝4.200:
l[1][7]:＝4.900:
l[1][8]:＝5.600:
l[1][9]:＝6.300:
＃左側の座標[2][x]
l[2][1]:＝－0.700:
l[2][2]:＝－1.400:
l[2][3]:＝－2.100:
l[2][4]:＝－2.800:
l[2][5]:＝－3.500:
l[2][6]:＝－4.200:
l[2][7]:＝－4.900:
l[2][8]:＝－5.600:
l[2][9]:＝－6.300:
＃直線表示
li[ATOM]:＝line([l[1][ATOM－1], 0, 0], [l[2][ATOM－1], 0, 0], thickness＝2, color＝black):
＃計算と結果の表示
print(行列は以下のようになります);
print(c);
＃固有値と固有ベクトルの計算 2 通り
Va, Vb:＝evalf(Eigenvectors(c)):
Vd:＝evalf(Eigenvectors(c, output＝'list')):
＃説明文
print(軌道のエネルギー);
＃実数部分の取り出し
```

```
＃固有値の複素数における実数部分の取り出し
for i from 1 to ATOM do
 Va1[i]:＝Re(Va[i]):
end do:
＃固有値の実数化
ATOM11:＝ATOM/2:
＃分数判定
ATOM12:＝type(ATOM11, fraction):
＃偶数のとき
if ATOM12＝false then
 ATOM21:＝ATOM11＋1:
＃奇数のとき
elif ATOM12＝true then
 ATOM21:＝ATOM11＋1/2:
end if:
for i from 1 to ATOM21 do
 Vd[i][1]:＝Re(Vd[i][1]):
end do:
＃昇順に並べ替え
Va2:＝sort([seq(Va 1[i], i＝1..ATOM)], `<`):
＃縦に表示
for i from 1 to ATOM do
 print(epsilon[ATOM－i＋1]＝alpha＋Va2[i]*beta);
end do:
＃軌道を変えるループ
orbitalloop:＝1:
while orbitalloop＝1 do
 ＃軌道の選択
 orbitalcnt:＝1:
 while orbitalcnt＝1 do
  print(軌道は合計，ATOM, あります);
print(何番目の軌道を表示しますか？);
＃軌道の入力
```

```
orbital:＝readstat("Please input orbit"):
＃軌道の判定
if orbital〈＝0 then
 print(指定の軌道は存在しません);
elif orbital〉ATOM then
    print(指定の軌道は存在しません);
else
    print(orbital, 番目の軌道を表示します);
    orbitalcnt:＝0:
end if:
 end do:
 ＃使用者の選択部分
 ＃固有値が一致するまで繰り返し
p:＝1:
while Va2[ATOM－orbital＋1]〈〉Vd[p][1]do
 p:＝p＋1:
end do:
＃固有値に対応する固有ベクトルの設定
＃実数化→二乗
for i from 1 to ATOM do
   Vd31[1][i]:＝Re(Vd[p][3][1][i]):
   Vd32[1][i]:＝Vd31[1][i]*Vd31[1][i]:
end do:
＃全ての二乗を合計し，平方根をとる
Vd41[1]:＝add(Vd32[1][i],i＝1..ATOM):
Vd42[1]:＝root(Vd41[1],2):
＃固有ベクトルを割る
for i from 1 to ATOM do
 EVC[1][i]:＝Vd31[1][i]/Vd 42[1]:
end do:
print(選択した軌道のエネルギーは);
print(epsilon[orbital]＝alpha＋Vd[p][1]*beta);
＃分子軌道
```

```
print(対応する分子軌道は次のようになります);
print(phi[orbital]＝add(EVC[1][i]*x[i], i＝1..ATOM));
 ＃色の設定
  Aodd:＝type(ATOM, odd):＃奇数判定
＃clr は color の略
＃cen は中央　l, r は左側，右側　u, d は up, down の略
＃奇数のとき
if Aodd＝true then
 for i from 1 to ATOM do
    if i＝(ATOM/2)＋(1/2)then
     if EVC[1][i]〉0 then
      clrcenu[3][i]:＝red:
      clrcend[3][i]:＝blue:
     elif EVC[1][i]〈0 then
      clrcenu[3][i]:＝blue:
    clrcend[3][i]:＝red:
   elif EVC[1][i]＝0 then
    clrcenu[3][i]:＝white:
      clrcend[3][i]:＝white:
     end if:
    elif i〈ATOM/2 then
     if EVC[1][i]〉0 then
     clrlu[2][i]:＝red:
     clrld[2][i]:＝blue:
    elif EVC[1][i]〈0 then
   clrlu[2][i]:＝blue:
   clrld[2][i]:＝red:
  elif EVC[1][i]＝0 then
     clrlu[2][i]:＝white:
     clrld[2][i]:＝white:
    end if:
   elif i〉(ATOM/2)＋1 then
   if EVC[1][i]〉0 then
```

```
        clrru[1][i−ATOM/2−1/2]:=red:
        clrrd[1][i−ATOM/2−1/2]:=blue:
       elif EVC[1][i]<0 then
      clrru[1][i−ATOM/2−1/2]:=blue:
        clrrd[1][i−ATOM/2−1/2]:=red:
     elif EVC[1][i]=0 then
        clrru[1][i−ATOM/2−1/2]:=white:
        clrrd[1][i−ATOM/2−1/2]:=white:
       end if:
      end if:
     end do:
＃偶数のとき
elif Aodd=false then
 for i from 1 to ATOM do
  if i<=ATOM/2 then
      if EVC[1][i]>0 then
       clrlu[2][i]:=red:
       clrld[2][i]:=blue:
      elif EVC[1][i]<0 then
       clrlu[2][i]:=blue:
       clrld[2][i]:=red:
      elif EVC[1][i]=0 then
       clrlu[2][i]:=white:
       clrld[2][i]:=white:
      end if:
     elif i>ATOM/2 then
      if EVC[1][i]>0 then
       clrru[1][i−ATOM/2]:=red:
       clrrd[1][i−ATOM/2]:=blue:
      elif EVC[1][i]<0 then
       clrru[1][i−ATOM/2]:=blue:
       clrrd[1][i−ATOM/2]:=red:
      elif EVC[1][i]=0 then
```

```
        clrru[1][i－ATOM/2]:＝white:
        clrrd[1][i－ATOM/2]:＝white:
       end if:
      end if:
     end do:
end if:
 ＃球体の設定
 ATOM1:＝ATOM/2:
＃分数判定
ATOM2:＝type(ATOM1, fraction):
＃中心線と描写の仕方
als:＝(li[ATOM], axes＝box, style＝HIDDEN):
＃原子数の判定
＃false→原子数が偶数
if ATOM2＝false then
 for i from 1 to ATOM do
  if i〈＝ATOM/2 then
splu[2][i]:＝sphere([l[2][ATOM − 2*i＋1], 0, abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝
clrlu[2][i]):
spld[2][i]:＝sphere([l[2][ATOM − 2*i＋1], 0, − abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝
clrld[2][i]):
    elif i〉ATOM/2 then
spru[1][i]:＝sphere([l[1][2*i－ATOM－1], 0, abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝
clrru[1][i－ATOM/2]):
sprd[1][i]:＝sphere([l[1][2*i−ATOM−1], 0, −abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝
clrrd[1][i－ATOM/2]):
    end if:
  end do:
image:＝display 3 d(seq([splu[2][i], spld[2][i]], i＝1..ATOM/2), seq([spru[1][i], sprd[1][i]],
i＝ATOM/2＋1..ATOM), als):
＃true→原子数が奇数
elif ATOM2＝true then
  for i from 1 to ATOM do
```

```
    if i＝(ATOM/2)＋1/2 then
spcenu[3][i]:＝sphere([0, 0, abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝clrcenu[3][i]):
spcend[3][i]:＝sphere([0, 0, －abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝clrcend[3][i]):
    elif i〈ATOM/2 then
splu[2][i]:＝sphere([1[2][ATOM － 2*i＋1], 0, abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝
clrlu[2][i]):
spld[2][i]:＝sphere([1[2][ATOM － 2*i＋1], 0, － abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝
clrld[2][i]):
    elif i〉(ATOM/2)＋1 then
spru[1][i]:＝sphere([1[1][2*i－ATOM－1], 0, abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝
clrru[1][i－ATOM/2－1/2]):
sprd[1][i]:＝sphere([1[1][2*i－ATOM－1], 0, －abs(EVC[1][i])/2], abs(EVC[1][i])/2, color＝
clrrd[1][i－ATOM/2－1/2]):
    end if:
end do:
image:＝display3d(seq([splu[2][i], spld[2][i]], i＝1..(ATOM/2)－1/2), seq([spru[1][i], sprd
[1][i]], i＝(ATOM/2)＋3/2..ATOM), spcenu[3][(ATOM＋1)/2], spcend[3][(ATOM＋1)/2],
als):
end if:
 ＃表示
 print(image);
 ＃軌道を変えるループ終了部分
 print(他の軌道も見ますか？);
orbitalloop1:＝1:
while orbitalloop1＝1 do
print(「1」他の軌道も見ます,「0」見ません);
orbitalloop1:＝readstat("Please input 0 or 1"):
if orbitalloop1＝1 then
   orbitalloop1:＝0:
 elif orbitalloop1＝0 then
   orbitalloop1:＝0:
   orbitalloop:＝0:
 else
```

```
  36
   print(他の数値が選ばれました);
   orbitalloop1:=1:
  end if:
end do:
end do:#終了設定
#最後の部分
print(他の分子も試してみますか？);
print(数値を入力してください);
fin1:=1:
print(「1」他も試します,「0」終了します);
while fin1=1 do
 ifend:=readstat("Please input 0 or 1");
#終了の選択
   if ifend=1 then
    fin1:=0:
   elif ifend=0 then
    fin1:=0:
    totalloop:=0:
   else
    print(他の数値が選ばれました);
   end if:
end do:
end do:
```

## 4. 発 展 性

将来, 炭素原子以外の異核原子を含む化合物への展開, さらに MOPAC と Maple, MATLAB と Maple の併用を基にした分子軌道法の応用教材作成を行う予定である。

### 引用文献

Maple を用いた量子化学入門教材作成　城西情報科学研究　第 16 巻　第 1 号（2006. 3）

### 参考図書

多数の成書があるが, 本化学科で教科書・参考書としている本を中心に記した。
（1） maple10 User Manual  Waterloo Maple Inc. 2005.

（2） ブラディー　一般化学（上・下）J. E. Brady et al，若山等訳，東京化学同人
（3） マクマリー　有機化学（上・中・下）伊東他訳，東京化学同人
（4） バーロー　物理化学　第 6 版，東京化学同人
（5） 大岩正芳著，初等量子化学　第二版，化学同人
（6） 福井謙一著，量子化学（近代工業化学 2），朝倉書店（1968）
（7） 米澤貞次郎，永田親善他，改定量子化学入門，化学同人

**（Received Feb. 2, 2007）**